



Learning non-metric visual similarity for image retrieval[☆]

Noa Garcia*, George Vogiatzis

Aston University, Birmingham B4 7ET, United Kingdom



ARTICLE INFO

Article history:

Received 16 April 2018

Received in revised form 12 January 2019

Accepted 19 January 2019

Available online 10 February 2019

Keywords:

Image retrieval

Visual similarity

Non-metric learning

ABSTRACT

Measuring visual similarity between two or more instances within a data distribution is a fundamental task in image retrieval. Theoretically, non-metric distances are able to generate a more complex and accurate similarity model than metric distances, provided that the non-linear data distribution is precisely captured by the system. In this work, we explore neural networks models for learning a non-metric similarity function for instance search. We argue that non-metric similarity functions based on neural networks can build a better model of human visual perception than standard metric distances. As our proposed similarity function is differentiable, we explore a *real* end-to-end trainable approach for image retrieval, i.e. we learn the weights from the input image pixels to the final similarity score. Experimental evaluation shows that non-metric similarity networks are able to learn visual similarities between images and improve performance on top of state-of-the-art image representations, boosting results in standard image retrieval datasets with respect standard metric distances.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

For humans, deciding whether two images are visually similar or not is, to some extent, a natural task. However, in computer vision, this is a challenging problem and algorithms do not always succeed in matching pictures that contain similar-looking elements. This is mainly because of the well-known *semantic gap* problem, which refers to the difference or gap between low-level image pixels and high-level semantic concepts. Estimating visual similarity is a fundamental task that seeks to break this semantic gap by accurately evaluating how alike two or more pictures are. Visual similarity is crucial for many computer vision areas including image retrieval, image classification and object recognition, among others.

Given a query image, content-based image retrieval systems rank pictures in a dataset according to how similar they are with respect to the input. This can be broken into two fundamental tasks: 1) computing meaningful image representations that capture the most salient visual information from pixels and 2) measuring accurate visual similarity between these image representations to rank images according to a similarity score.

In the last years, several methods to represent visual information from raw pixels in images have been proposed, first by designing

handcrafted features such as SIFT [1], then by compacting these local features into a single global image descriptor using different techniques such as Fisher Vectors [2] and more recently by extracting deep image representations from neural networks [3]. However, once two images are described by feature vectors, visual similarity is commonly measured by computing a standard metric between them. Although regular distance metrics, such as Euclidean distance or cosine similarity, are fast and easy to implement, they do not take into account the possible interdependency within the dataset, which means that even if a strong nonlinear data dependency is occurring in the visual collection, they might not be able to capture it. This suggests that learning a similarity estimation directly from visual data can improve the performance on image retrieval tasks, provided that the likely nonlinearity dependencies within the dataset are precisely learned by the similarity function.

In this work, we propose a model to learn a non-metric visual similarity function on top of image representations for pushing accuracy in image retrieval tasks. This idea is shown in Fig. 1. As in standard image retrieval systems, we extract K -dimensional visual vectors from images by using a convolutional neural network (CNN) architecture. Then, a *visual similarity neural network* is used to estimate the similarity score between a pair of images. Note that in standard systems this score is usually computed with a metric distance.

We design a supervised regression learning protocol so that different similarity degrees between images are precisely captured. Then, we directly apply the output of the model as a similarity estimation to rank images accordingly. In this way, the similarity network can be

[☆] This paper has been recommended for acceptance by Sinisa Todorovic, PhD.

* Corresponding author.

E-mail addresses: garciadn@aston.ac.uk (N. Garcia),

g.vogiatzis@aston.ac.uk (G. Vogiatzis).

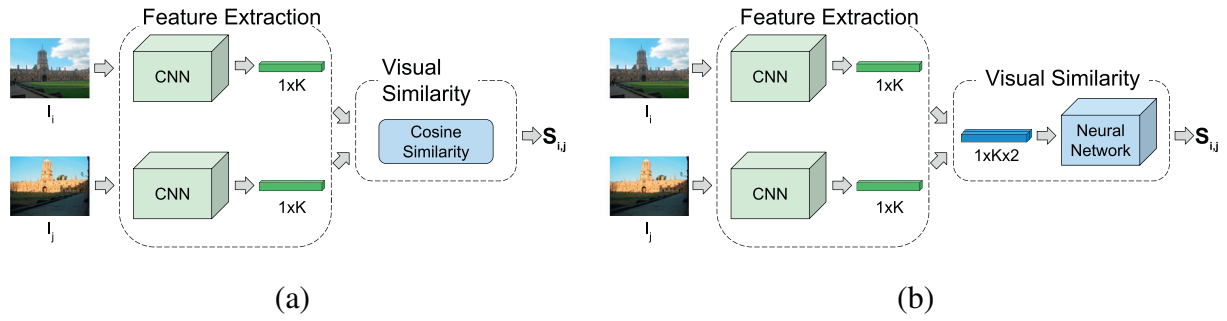


Fig. 1. Retrieval system based on metric distances versus our proposed model. (a) Standard systems use a metric distance function to estimate the visual similarity between a pair of feature vectors obtained from a feature extraction model. (b) Our proposed model estimates the similarity score, S_{ij} , from a pair of visual vectors by using a non-metric similarity network.

seen as a replacement of the standard metric distance computation, being able to mathematically fit visual human perception better than standard metrics and to improve results on top of them. The proposed similarity network is end-to-end differentiable, which allows us to build an architecture for *real* end-to-end training: from the input image pixels to the final similarity score. Experimental evaluation shows that performance on standard image retrieval datasets is boosted when the similarity function is directly learnt from the visual data instead of using a rigid metric distance.

Many techniques have been envisaged to boost image retrieval performance in the past, such as query expansion and re-ranking [4], network fine-tuning [3,5] or feature fusion [6,7]. However, these techniques are not competitors of our method but optional add-ons, as we argue that the methodology proposed in this work can be applied along with all of them in the same way as they are being applied on systems based on metric distances. Moreover, training a similarity network as we propose is computationally simpler than fine-tuning the whole feature representation network (i.e. network fine-tuning), extracting multiple features using different networks (i.e. feature fusion) or computing multiple queries per image (i.e. query expansion and re-ranking).

The main contributions of this work are summarised as follows:

1. We present a neural network architecture to model visual similarities, which introduces a new and simple method to boost performance in image retrieval by only training the last stage of the system.
2. We propose a novel regression loss function specifically designed for improving similarity scores on top of standard metrics in image retrieval tasks.
3. We design a *real* end-to-end system for content-based image retrieval that can be trained from the input image pixels to the final similarity score.
4. We empirically show the efficacy of our method in standard image retrieval datasets. Via our ablation study, we show that the proposed system can successfully compute visual similarities on top of different standard retrieval features, outperforming cosine similarity and metric learning in most of the datasets.

This paper is structured as follows: a revision of relevant work can be found in Section 2; our method is detailed in Section 3; experimental evaluation is described in Section 4; and conclusions are stated in Section 5.

2. Related work

In this section relevant work in image retrieval and similarity learning is carefully reviewed.

2.1. Content-based image retrieval

Content-based image retrieval searches for images by considering their visual content. Given a query image, pictures in a collection are ranked according to their visual similarity with respect to the query. Early methods represent the visual content of images by a set of hand-crafted features, such as SIFT [1]. As a single image may contain hundreds of these features, aggregation techniques like bag-of-words (BOW) [8], Fisher Vectors [2] or VLAD [9] encode local descriptors into a compact vector, thereby improving computational efficiency and scalability. More recently, because of the latest advancements on deep learning, features obtained from convolutional neural networks (CNN) have rapidly become the new state-of-the-art in image retrieval.

2.2. Deep learning for image retrieval

Deep image retrieval extracts activations from CNNs as image representations. At first, some methods [3,10–12] proposed to use representations from one of the last fully connected layers of networks pre-trained on the classification ImageNet dataset [13]. When deeper networks such as GoogLeNet [14] and VGG [15] appeared, some authors [10,16–18] showed that mid-layer representations obtained from the convolutional layers performed better in the retrieval task. Since then, there have been several attempts to aggregate these high-dimensional convolutional representations into a compact vector. For example, [19,17] compacted deep features by using VLAD, [20] encoded the neural codes into a histogram of words, [16,21] applied sum-pooling to obtain a compact representation and [22,23] aggregated deep features by max-pooling them into a new vector. A different approach is to train the network to directly learn compact binary codes end-to-end [24,25]. Some authors have shown that fine-tuning the networks with similar data to the target task increases the performance significantly [3,5,26–28]. Finally, recent work has shown that adding attention models to select meaningful features can be also beneficial for image retrieval [29,30].

All of these methods are focused on finding high quality features to represent visual content efficiently and visual similarity is computed by simply applying a standard metric distance. General metrics, such as Euclidean distance or cosine similarity, however, might be failing to consider the inner data structure of these visual representations. Learning a similarity function directly from data may help to capture the human perception of visual similarity in a better way.

2.3. Similarity learning

Some of the most popular similarity learning work, such as OASIS [31] and MLR [32], are based on linear metric learning by optimizing the weights of a linear transformation matrix. For example,

Yang et al. [33] proposed a framework for ranking elements in retrieval tasks by solving a linear optimization problem. Although linear methods are easier to optimize and less prone to overfitting, nonlinear algorithms are expected to achieve higher accuracy modeling the possible nonlinearities of data.

Nonlinear similarity learning based on deep learning has been recently applied to many different visual contexts. In low-level image matching, CNNs have been trained to match pairs of patches for stereo matching [34,35] and optical flow [36,37]. In high-level image matching, deep learning techniques have been proposed to learn low-dimensional embedding spaces in face verification [38], retrieval [39,40], classification [41–43] and product search [44], either by using siamese [38] or triplet [40] architectures. More recently, deep similarity learning has also been applied to fabric image retrieval [45] by using triplets of samples to ensure that similar features are mapped closer than non-similar features.

In general, these methods rely on learning a mapping from image pixels to a low dimensional target space to compute the final similarity decision by using a standard metric. They are designed to find the best projection in which a linear distance can be successfully applied. Instead of projecting the visual data into some linear space, that may or may not exist, our approach seeks to learn the non-metric visual similarity score itself. Similarly, [46] and [47] used a CNN to decide whether or not two input images are a match, applied to pedestrian re-identification and patch matching, respectively. In these methods, the networks are trained as a binary classification problem (i.e. same or different pedestrian/patch), whereas in an image retrieval ranking problem, a regression score is required. Inspired by the results of [11], which showed that combining deep features with similarity learning techniques can be very beneficial for the performance of image retrieval systems, we propose to train a deep learning algorithm to learn non-metric similarities for image retrieval and improve results in top of high quality image representation methods.

Neural networks have been previously proposed to model relationships between objects in different domains, such as classification in few-shot learning [48,49] or visual question answering [50]. The main difference between these methods and this work lies in the optimization problem to be solved: whereas [48–50] aim to learn if a certain relation between a pair of images is occurring by optimizing a classification loss function (e.g. whether a query image is similar to samples from a known class or not), we introduce a novel loss function specifically designed to solve ranking problems, which improves similarity scores on top of a standard metric by returning a regression value (i.e. how similar the two images are).

3. Methodology

In this section we present our proposed method to learn a non-metric visual similarity function from the visual data distribution for image retrieval.

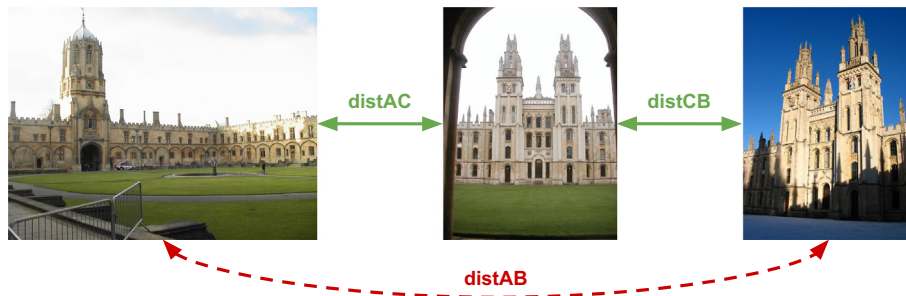


Fig. 2. Triangle inequality in image retrieval. Triangle inequality does not fit human visual perception, where distAB is expected to be bigger than distAC plus distCB .

3.1. Visual similarity

Visual similarity measures how alike two images are. Formally, given a pair of images I_i and I_j in a collection of images ξ , we define s_{ij} as their similarity score. The higher s_{ij} is, the more similar I_i and I_j are. To compute s_{ij} , images are represented by K -dimensional image representations, which are obtained by mapping image pixels into the feature space \mathbb{R}^K , as $x_k = f(I_k, w_f)$ with $I_k \in \xi$, where $f(\cdot)$ is a non-linear image representation function and w_f its parameters. We propose to learn a visual similarity function, $g(\cdot)$, that maps a pair of image representations x_i and x_j into a visual score as:

$$s_{ij} = g(f(I_i, w_f), f(I_j, w_f), w_g) \\ \text{s.t. } s_{ij} > s_{i,k} \rightarrow I_i, I_j \text{ more similar than } I_i, I_k \quad (1)$$

with $I_i, I_j, I_k \in \xi$ and w_g being the trainable parameters of the similarity function.

Visual similarity functions are commonly based on metric distance functions such as $g(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$ or $g(x_i, x_j) = \|x_i - x_j\|$, i.e. cosine similarity and Euclidean distance, respectively. Metric distance functions, $d(\cdot)$, perform mathematical comparisons between pairs of objects in a collection Π , by satisfying the following axioms:

1. $d(a, b) \geq 0$ (non-negativity)
2. $d(a, b) = 0 \Leftrightarrow a = b$ (identity)
3. $d(a, b) = d(b, a)$ (symmetry)
4. $d(a, b) \leq d(a, c) + d(c, b)$ (triangle inequality)

with $\forall a, b, c \in \Pi$.

However, metric axioms are not always the best method to represent visual human perception [51–53]. For example, non-negative and identity axioms are not required in visual perception as long as relative similarity distances are maintained. Symmetry axiom is not always true, as human similarity may be influenced by the order of the objects being compared. Finally, triangle inequality does not correspond to visual human perception either. It can be easily seen when considering the images of a person, a horse and a centaur: although a centaur might be visually similar to both a person and a horse, the person and the horse are not similar to each other. A visual example applied to image retrieval can be seen in Fig. 2.

For a better mathematical representation of the visual human perception, we propose to learn a non-metric visual similarity function without requiring to satisfy the rigid metric axioms.

3.2. Similarity network

To fit visual human perception better than metric distance functions, we propose to learn a similarity function from the visual data using neural networks. This similarity network is composed of

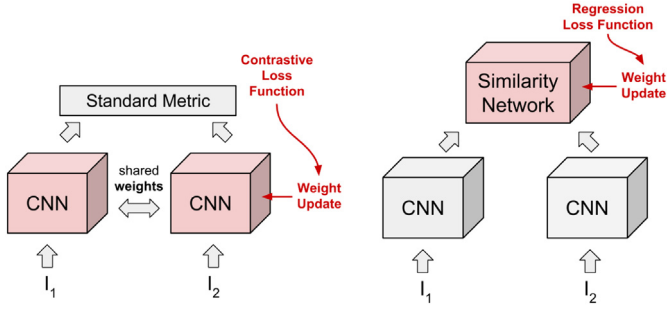


Fig. 3. Similarity versus Siamese networks. Siamese networks (left) learn to map pixels into vector representations, whereas Similarity networks (right) learn a similarity function on top of the vector representations.

a set of fully connected layers, each one of them, except by the last one, followed by a ReLU [54] non-linearity.

The input of the network is a concatenated pair of image representations vectors, x_i and x_j , which can be obtained using any standard technique, such as [3,23] or any other. The output is a similarity score, s_{ij} . In that way, the similarity network learns the similarity function, $g(\cdot)$, from the image representation vectors and replaces the metric distance function used in standard systems.

At this point, we would like to note that the proposed similarity network is conceptually different to the siamese architecture in [38], as shown in Fig. 3. Siamese networks use pairs of images to learn the feature extraction function, $f(\cdot)$, which maps image pixels into vector representations. Then, similarity is still computed with a metric distance function, such as cosine similarity or Euclidean distance. In contrast, our approach learns the function $g(\cdot)$ on top of the image representations, replacing the standard metric distance computation.

3.3. Similarity training

We design the training of the similarity network as a supervised regression task. However, as providing similarity labels for every possible pair of training images is infeasible, we propose a training procedure in which the visual similarity is learned progressively using standard image classification annotations. The model is trained to discriminate whether two images, I_i, I_j , are similar or dissimilar. Then, a similarity score, s_{ij} , is assigned accordingly by improving a standard similarity function, $sim(\cdot)$. To optimize the weights, w_g , of the similarity function $g(\cdot)$ from Eq. (1), the following regression loss function is computed between each training pair of image representations, x_i, x_j :

$$\mathcal{L}(I_i, I_j) = |s_{ij} - \ell_{ij}(sim(x_i, x_j) + \Delta) - (1 - \ell_{ij})(sim(x_i, x_j) - \Delta)| \quad (2)$$

where Δ is a margin parameter and ℓ_{ij} is defined as:

$$\ell_{ij} = \begin{cases} 1 & \text{if } I_i \text{ and } I_j \text{ are similar} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In other words, the similarity network learns to increase the similarity score when two matching images are given and to decrease it when a pair of images is not a match. Similarity between pairs might be decided using different techniques, such as image classes, score based on local features or manual labeling, among others [55]. Without loss of generality, we consider two images as similar when they belong to the same annotated class and as dissimilar when they belong to different classes.

Choosing appropriate examples when using pairs or triplets of samples in the training process is crucial for a successful training [26,27,56]. This is because if the network is only trained by using *easy pairs* (e.g. a car and a dog), it will not be able to discriminate between *difficult pairs* (e.g. a car and a van). We design our training protocol by emphasizing the training of difficult examples. First, we randomly select an even number of similar and dissimilar pairs of training samples and train the similarity network until convergence. We then choose a new random set of images and compute the similarity score between all possible pairs by using the converged network. Pairs in which the network output is worse than the metric distance function measure are selected as difficult pairs for retraining, where a worse score means a score that is lower in the case of a match and higher in the case of a non-match. Finally, the difficult pairs are added to the training process and the network is trained until convergence one more time. Examples of difficult image pairs are shown in Fig. 4.

4. Experimental evaluation

In this section, we present the experimental evaluation we perform to validate the proposed non-metric similarity network.

4.1. Image retrieval datasets

Our approach is evaluated on three standard image retrieval datasets: OXFORD5K [57], PARIS6K [58] and LAND5K, a validation subset of LANDMARKS [3] dataset. OXFORD5K consists on 5062 images of 11 different Oxford landmarks and 55 query images. PARIS6K contains 6412 images of 11 different Paris landmarks and 55 queries. LAND5K consists of 4915 images from 529 classes with a random selection of 45 images to be used as queries. For experiments on larger datasets, we also use the standard large-scale versions OXFORD105K and PARIS106K, by including 100,000 distractor images [57]. In both OXFORD5K and PARIS6K collections query images are cropped according to the region of interest and evaluation is performed by computing the mean Average Precision (mAP). For LAND5K results are also reported as mAP, considering an image to be relevant to a query when they both belong to the same class.



Fig. 4. Example of difficult pairs. Dissimilar images with lower score than the metric distance (left) and similar images with higher score than the metric distance (right).

Table 1

Architecture discussion. Fully connected layers are denoted as (FC-{filters}).

Config	Architecture	Params	MSE	ρ
A	FC-1024, FC-1024, FC-1	2.1 M	0.00035	0.909
B	FC-4096, FC-4096, FC-1	21 M	0.00019	0.965
C	FC-8192, FC-8192, FC-1	76 M	0.00012	0.974
D	FC-4096, FC-4096, FC-4096, FC-1	38 M	0.00019	0.964

For training, we use the cleaned version of the LANDMARKS [3] dataset from [26]. Due to broken URLs, we could only download 33,119 images for training and 4915 for validation. To ensure visual similarity is learnt from relevant data, we create two more training sets, named LANDMARKS-EXTRA500 and LANDMARKS-EXTRA, by randomly adding about 500 and 2000 images from OXFORD5K and PARIS6K classes to LANDMARKS, respectively. Query images are not added in any case and they remain unseen by the system.

4.2. Experimental details

4.2.1. Image representation

Unless otherwise stated, we use RMAC [23] as image representation method. VGG16 network is used off-the-shelf without any retraining or fine-tuning. Images are re-scaled up to 1024 pixels, keeping their original aspect ratio. RMAC features are very sensitive to the PCA matrices used for normalization. For consistency, we use the PCA whitening matrices trained on PARIS5K on all the datasets, instead of using different matrices in each testing collection. This leads to slightly worse results than the ones provided in the original paper.

4.2.2. Similarity training

We use cosine similarity, $\text{sim}(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$, as the similarity function in Eq. (2). For a faster convergence, we warm-up the weights of the similarity network by training it with random generated pairs of vectors and $\Delta = 0$. In this way, the network first learns to imitate the cosine similarity. Visual similarity is then trained using almost a million of image pairs. We experiment with several values of the margin parameter Δ , ranging from 0.2 to 0.8. The network is optimized using backpropagation and stochastic gradient descent with a learning rate of 0.001, a batch size of 100, a weight decay of 0.0005 and a momentum of 0.9.

4.2.3. Computational cost

Standard metric functions are relatively fast and computationally cheap. Our visual similarity network involves the use of millions of parameters that inevitably increase the computational cost. However, it is still feasible to compute the similarity score in a reasonable amount of time. In our experiments, training time is about 5 h in a GeForce GTX 1080 GPU without weight warm-up and testing time for a pair of images is 1.25 ms on average (0.35 ms when using cosine similarity in CPU).

4.3. Architecture discussion

We first experiment with four different architectures (Table 1) and compare the performance of each configuration during the network warm-up ($\Delta = 0$), by using 22.5 million and 7.5 million pairs of randomly generated vectors for training and validation, respectively. During the training warm-up, the network is intended to imitate the cosine similarity. We evaluate each architecture by computing the mean squared error, MSE, and the correlation coefficient, ρ , between the network output and the cosine similarity. Configuration C, which is the network with larger number of parameters, achieves the best MSE and ρ results. Considering a trade-off between performance and number of parameters of each architecture, we keep configuration B as our default architecture for the rest of the experiments.

4.4. Similarity evaluation

We then study the benefits of using a non-metric similarity network for image retrieval by comparing it to several similarity methods. RMAC [23] is used as image representation method in all the experiments. Similarity functions under evaluation are:

- Cosine: the similarity between a pair of vectors is computed with the cosine similarity: $\text{cosine}(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$. No training is required.
- OASIS: well-established OASIS algorithm [31] is used to learn a linear function to map a pair of vectors into a similarity score. The training of the matrix transformation is performed in a supervised way by providing the class of each image.
- Linear: we learn an affine transformation matrix to map a pair of vectors into a similarity score by optimizing Eq. (2) in a supervised training. Classes of images are provided during training. The margin Δ is set to 0.2.
- SimNet, SimNet*: the similarity function is learnt with our proposed similarity network by optimizing Eq. (2) with (SimNet*) or without (SimNet) difficult pairs refinement. Classes of images are provided during training and different margin Δ are tested, ranging from 0.2 to 0.8.

Results are summarised in Table 2. Trained similarity networks (SimNet, SimNet*) outperform trained linear methods (OASIS, Linear) in all but one testing datasets. As all Linear, SimNet and SimNet* are trained using the same supervised learning protocol and images, the results suggest that the improvement obtained in our method is not because of the supervision but because of the non-metric nature of the model, which is supposed to fit the human visual perception more accurately. Moreover, when using LANDMARKS-EXTRA as training dataset, results are boosted with respect to the standard metric, achieving improvements ranging from 20% (OXFORD5K) to 40% (PAIRS6K). When using LANDMARKS-EXTRA500 dataset, our similarity networks also improve mAP with respect to the cosine similarity in

Table 2Similarity evaluation. mAP when using different similarity functions. Δ value is set in brackets.

	LANDMARKS			LANDMARKS-EXTRA500			LANDMARKS-EXTRA		
	Ox5k	Pa6k	La5k	Ox5k	Pa6k	La5k	Ox5k	Pa6k	La5k
Cosine	0.665	0.638	0.564	0.665	0.638	0.564	0.665	0.638	0.564
OASIS	0.514	0.385	0.578	0.570	0.651	0.589	0.619	0.853	0.579
Linear (0.2)	0.598	0.660	0.508	0.611	0.632	0.514	0.602	0.581	0.502
SimNet (0.2)	0.658	0.460	0.669	0.717	0.654	0.671	0.718	0.757	0.668
SimNet* (0.2)	0.655	0.503	0.697	0.719	0.677	0.693	0.786	0.860	0.662
SimNet* (0.4)	0.637	0.504	0.737	0.703	0.701	0.745	0.794	0.878	0.706
SimNet* (0.6)	0.613	0.514	0.776	0.703	0.716	0.776	0.789	0.885	0.735
SimNet* (0.8)	0.600	0.511	0.783	0.685	0.710	0.803	0.808	0.891	0.758

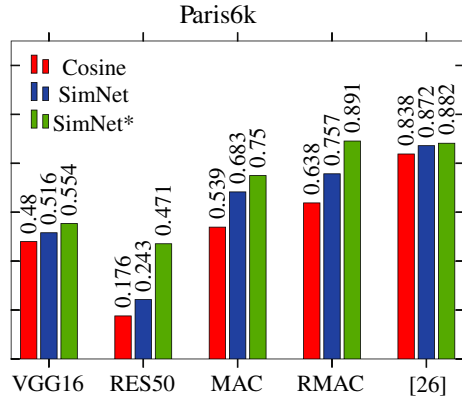


Fig. 5. Image representation discussion. mAP for different visual similarity techniques on top of different feature extraction methods.

the three testing datasets. This indicates that visual similarity can be learnt even when using a reduced subset of the target image domain. However, visual similarity does not transfer well across domains when no images of the target domain are used during training, which is a well-known problem in metric learning systems [59]. In that case, cosine similarity is the best option over all the methods.

4.5. Image representation discussion

Next, we study the generalisation of our similarity networks when used on top of different feature extraction methods: the output of a VGG16 network [15], the output of a ResNet50 network [60], MAC [23], RMAC [23] and the model from [27]. We compare the results of our networks, SimNet and SimNet*, against cosine similarity. Results

Table 3

End-to-end training. mAP when fine-tuning different parts of the pipeline. In grey, the modules that are fine-tuned in every experiment.

Features	Sim.	Ox5k	PA6K	LA5K
MAC	Cosine	0.481	0.539	0.494
MAC	SimNet	0.509	0.683	0.589
MAC	SimNet	0.555	0.710	0.685

are provided in Fig. 5. Our similarity networks outperform cosine similarity in all the experiments, improving retrieval results when used on top of any standard feature extraction method. Moreover, performance is boosted when SimNet* is applied, specially in features with poor retrieval performance, such as ResNets.

4.6. Domain adaptation

We further investigate the influence of the training dataset on the similarity score when it is transferred between different domains or collections of images. As already noted in Table 2, visual similarity does not transfer well across domains, and a subset of samples from the target dataset is required during training to learn a meaningful similarity function. This is mainly because similarity estimation is a problem-dependent task, as the similarity between a pair of elements depends on the data collection. Thus, in Fig. 6, we explore the effect on performance when we use different subsets of samples from the target collection in addition to the LANDMARKS dataset. To add relevant samples progressively to the training set, we assign a class label to each image in the OXFORD5K and PARIS6K collections. These datasets do not provide class labels *per se*, so to overcome this issue, we use the file name of each image as its class label.

There is a clear correlation between the similarity network performance and the number of samples from the target dataset used

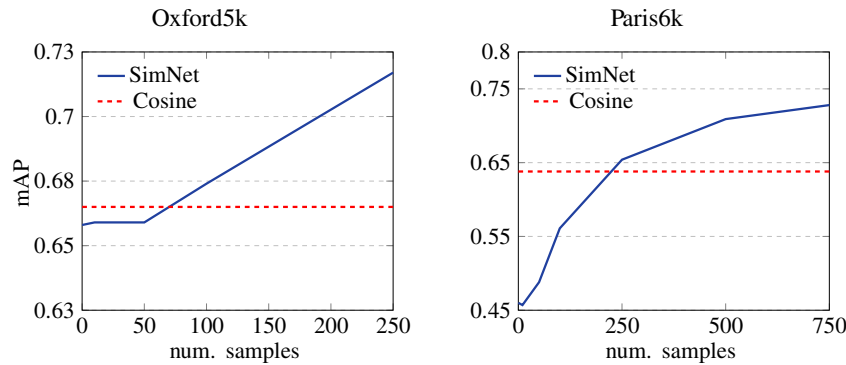


Fig. 6. Domain adaptation mAP when using different number of target samples in the training set.

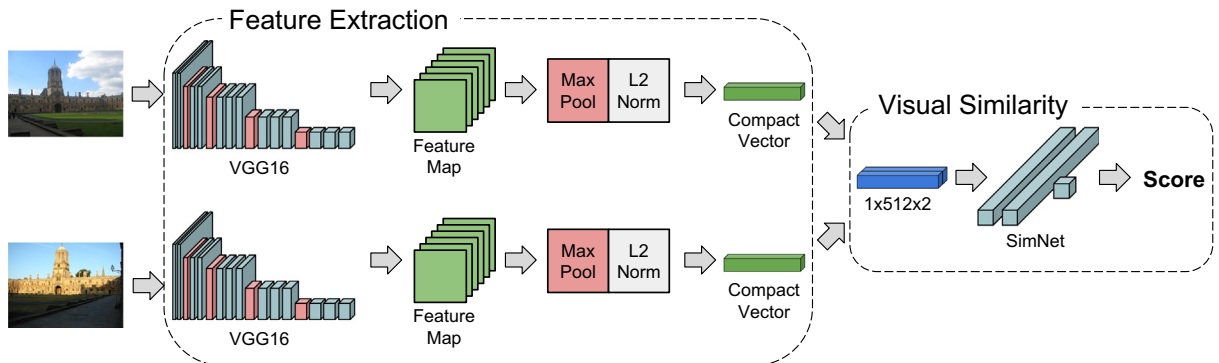


Fig. 7. End-to-end model. MAC as feature extraction and SimNet as visual similarity.

Table 4

State-of-the-art comparison (off-the-shelf). Dim corresponds to the dimensionality of the feature representation and Similarity is the similarity function.

Method	Dim	Similarity	Ox5k	Ox105k	PA6k	PA106k
Babenko et al. [3]	512	L2	0.435	0.392	–	–
Razavian et al. [10]	4096	Averaged L2	0.322	–	0.495	–
Wan et al. [11]	4096	OASIS	0.466	–	0.867	–
Babenko et al. [16]	256	Cosine	0.657	0.642	–	–
Yue et al. [17]	128	L2	0.593	–	0.59	–
Kalantidis et al. [21]	512	L2	0.708	0.653	0.797	0.722
Mohedano et al. [20]	25k	Cosine	0.739	0.593	0.82	0.648
Salvador et al. [28]	512	Cosine	0.588	–	0.656	–
Tolias et al. [23]	512	Cosine	0.669	0.616	0.83	0.757
Jimenez et al. [29]	512	Cosine	0.712	0.672	0.805	0.733
Ours ($\Delta = 0.8$)	512	SimNet*	0.808	0.772	0.891	0.818

Table 5

State-of-the-art comparison (fine-tuned). Dim corresponds to the dimensionality of the feature representation and Similarity is the similarity function.

Method	Dim	Similarity	Ox5k	Ox105k	PA6k	PA106k
Babenko et al. [3]	512	L2	0.557	0.522	–	–
Gordo et al. [26]	512	Cosine	0.831	0.786	0.871	0.797
Wan et al. [11]	4096	OASIS	0.783	–	0.947	–
Radenovic et al. [27]	512	Cosine	0.77	0.692	0.838	0.764
Salvador et al. [28]	512	Cosine	0.71	–	0.798	–
Gordo et al. [5]	2048	Cosine	0.861	0.828	0.945	0.906
Ours ($\Delta = 0.8$)	512	SimNet*	0.882	0.821	0.882	0.829

during training. Indeed, in agreement with previous work in metric learning [59], we observe that not considering samples from the target dataset to train a similarity function might be harmful. The similarity network, however, outperforms standard metric results even when a small number of samples from the target collection is used during training: only 100 images from Oxford5k and 250 images from Paris6k are required to outperform cosine similarity in OXFORD5K and PARIS6K datasets, respectively. This suggests that the similarity network is able to generalise from a small subset of target samples, instead of memorising distances in the training collection.

4.7. End-to-end training

So far, we have isolated the similarity computation part to verify that the improvement in the testing datasets compared to when using other similarity methods is, in fact, due to the visual similarity network. However, with all the modules of the retrieval system being differentiable, an end-to-end training model is also possible. End-to-end methods have been shown to achieve outstanding results in many different problems, including the aggregation of video features [61], stereo matching [47], person re-identification [46] or self-driving cars [62]. In image retrieval, however, only end-to-end methods for learning the feature representations are proposed [5,27], leaving the final similarity score to be computed with a cosine similarity. In this section we explore a real end-to-end training architecture for image retrieval, which is presented in Fig. 7.

For the feature extraction part, we adopt MAC [23], although any differentiable image representation method can be used. To obtain MAC vectors, images are fed into a VGG16 network [15]. The output of the last convolutional layer is max-pooled and l2-normalized. For the visual similarity part, we use SimNet with $\Delta = 0.2$. As the whole architecture is end-to-end differentiable, the weights are fine-tuned through backpropagation. We first train the similarity network by freezing the VGG16 weights. Then, we unfreeze all the layers and fine-tune the model one last time. As all the layers have been already pre-trained, the final end-to-end fine-tuning is performed in only about 200,000 pairs of images from LANDMARKS-EXTRA dataset for just 5000 iterations.

Results are provided in Table 3. There is a significant improvement when using the similarity network instead of the cosine similarity, as

already seen in the previous section. When the architecture is trained end-to-end results are improved even more, since fine-tuning the entire architecture allows a better fit to a particular dataset.

4.8. State-of-the-art comparison

Finally, we compare our method against several state-of-the-art techniques. As standard practice, works are split into two groups: off-the-shelf and fine-tuned. Off-the-shelf are techniques that extract image representations by using pre-trained CNNs, whereas fine-tuned methods retrain the network parameters with a relevant dataset to compute more accurate visual representation. For a fair comparison, we only consider methods that represent each image with a single visual vector, without query expansion or image re-ranking.

Off-the-shelf results are shown in Table 4 and fine-tuned results are presented in Table 5. When using off-the-shelf RMAC features, our SimNet* approach outperforms previous methods in every dataset. To compare against fine-tuned methods, we compute RMAC vectors using the fine-tuned version of VGG16 proposed in [27]. Accuracy is boosted when our similarity network is used instead of the analogous cosine similarity method [27]. SimNet* achieves the best mAP precision in OXFORD5K dataset and comes second in OXFORD105K and PARIS106K after [5], which uses the more complex and higher-dimensional ResNet network [60] for image representation.

5. Conclusions

We have presented a method for learning visual similarity directly from visual data. Instead of using a metric distance function, we propose to train a neural network model to learn a similarity score between a pair of visual representations. Our method is able to capture visual similarity better than other techniques, mostly because of its non-metric nature. As all the layers in the similarity network are differentiable, we also propose an end-to-end trainable architecture for image retrieval. Experiments on standard collections show that results are considerably improved when a similarity network is used. Finally, our work can push performance in image retrieval systems on top of high-quality image features, while it can still be applied with query expansion or image re-ranking methods.

References

- [1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* 60 (2) (2004) 91–110.
- [2] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 3384–3391.
- [3] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, *European Conference on Computer Vision*, Springer, 2014, pp. 584–599.
- [4] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, *Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2911–2918.
- [5] A. Gordo, J. Almazan, J. Revaud, D. Larlus, End-to-end learning of deep visual representations for image retrieval, *Int. J. Comput. Vis.* 124 (2) (2017) 237–254.
- [6] X. Dong, Y. Yan, M. Tan, Y. Yang, I.W. Tsang, Late fusion via subspace search with consistency preservation, *IEEE Trans. Image Process.* 28 (1) (2018) 518–528.
- [7] X. Dong, L. Zheng, F. Ma, Y. Yang, D. Meng, Few-example object detection with model communication, *IEEE Trans. Pattern Anal. Mach. Intell.* (1) (2018) 1–1.
- [8] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, *International Conference on Computer Vision*, IEEE, 2003, pp. 1470.
- [9] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (9) (2012) 1704–1716.
- [10] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.
- [11] J. Wan, D. Wang, S.C.H. Hoi, P. Wu, J. Zhu, Y. Zhang, J. Li, Deep learning for content-based image retrieval: a comprehensive study, *ACM International Conference on Multimedia*, ACM, 2014, pp. 157–166.
- [12] Y. Liu, Y. Guo, S. Wu, M.S. Lew, DeepIndex for accurate and efficient image retrieval, *ACM on International Conference on Multimedia Retrieval*, ACM, 2015, pp. 43–50.
- [13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge, *Int. J. Comput. Vis.* 115 (3) (2015) 211–252.
- [14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
- [15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *International Conference on Learning Representations*, 2015.
- [16] A. Babenko, V. Lempitsky, Aggregating local deep features for image retrieval, *IEEE international conference on computer vision*, 2015, pp. 1269–1277.
- [17] J. Yue-Hei Ng, F. Yang, L.S. Davis, Exploiting local features from deep networks for image retrieval, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 53–61.
- [18] L. Xie, R. Hong, B. Zhang, Q. Tian, Image classification and retrieval are one, *ACM on International Conference on Multimedia Retrieval*, ACM, 2015, pp. 3–10.
- [19] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, *European Conference on Computer Vision*, Springer, 2014, pp. 392–407.
- [20] E. Mohamedano, K. McGuinness, N.E. O'Connor, A. Salvador, F. Marqués, X. Giro-i Nieto, Bags of local convolutional features for scalable instance search, *ACM on International Conference on Multimedia Retrieval*, ACM, 2016, pp. 327–331.
- [21] Y. Kalantidis, C. Mellina, S. Osindero, Cross-dimensional weighting for aggregated deep convolutional features, *European Conference on Computer Vision*, Springer, 2016, pp. 685–701.
- [22] A.S. Razavian, J. Sullivan, S. Carlsson, A. Maki, Visual instance retrieval with deep convolutional networks, *ITE Trans. Media Technol. Appl.* 4 (3) (2016) 251–258.
- [23] G. Toliás, R. Sircé, H. Jégou, Particular object retrieval with integral max-pooling of CNN activations, *International Conference on Learning Representations*, 2016.
- [24] V. Erin Liong, J. Lu, G. Wang, P. Moulin, J. Zhou, Deep hashing for compact binary codes learning, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2475–2483.
- [25] K. Lin, H.-F. Yang, J.-H. Hsiao, C.-S. Chen, Deep learning of binary hash codes for fast image retrieval, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 27–35.
- [26] A. Gordo, J. Almazan, J. Revaud, D. Larlus, Deep image retrieval: learning global representations for image search, *European Conference on Computer Vision*, Springer, 2016, pp. 241–257.
- [27] F. Radenović, G. Toliás, O. Chum, CNN image retrieval learns from BoW: unsupervised fine-tuning with hard examples, *European Conference on Computer Vision*, Springer, 2016, pp. 3–20.
- [28] A. Salvador, X. Giro-i Nieto, F. Marqués, S. Satoh, Faster R-CNN features for instance search, *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 9–16.
- [29] A. Jiménez, J.M. Alvarez, X. Giro Nieto, Class-weighted convolutional features for visual instance search, *British Machine Vision Conference*, 2017, pp. 1–12.
- [30] H. Noh, A. Araujo, J. Sim, T. Weyand, B. Han, Largescale image retrieval with attentive deep local features, *IEEE International Conference on Computer Vision*, 2017, pp. 3456–3465.
- [31] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking, *J. Mach. Learn. Res.* 11 (2010) 1109–1135.
- [32] B. McFee, G.R. Lanckriet, Metric learning to rank, *International Conference on Machine Learning*, 2010, pp. 775–782.
- [33] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, Y. Pan, A multimedia retrieval framework based on semi-supervised ranking and relevance feedback, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (4) (2012) 723–742.
- [34] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4353–4361.
- [35] W. Luo, A.G. Schwing, R. Urtasun, Efficient deep learning for stereo matching, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695–5703.
- [36] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, FlowNet: learning optical flow with convolutional networks, *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.
- [37] J. Thewlis, S. Zheng, P. Torr, A. Vedaldi, Fully-trainable deep matching, *Proceedings of the British Machine Vision Conference*, 2016.
- [38] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, *Computer Vision and Pattern Recognition*, 1, IEEE, 2005, pp. 539–546.
- [39] P. Wu, S.C. Hoi, H. Xia, P. Zhao, D. Wang, C. Miao, Online multimodal deep similarity learning with application to image retrieval, *ACM international conference on Multimedia*, ACM, 2013, pp. 153–162.
- [40] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.
- [41] E. Hoffer, N. Ailon, Deep metric learning using triplet network, *International Workshop on Similarity-Based Pattern Recognition*, Springer, 2015, pp. 84–92.
- [42] Q. Qian, R. Jin, S. Zhu, Y. Lin, Fine-grained visual categorization via multi-stage metric learning, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3716–3724.
- [43] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4004–4012.
- [44] S. Bell, K. Bala, Learning visual similarity for product design with convolutional neural networks, *ACM Trans. Graph.* 34 (4) (2015) 98.
- [45] D. Deng, R. Wang, H. Wu, H. He, Q. Li, X. Luo, Learning deep similarity models with focus ranking for fabric image retrieval, *Image Vis. Comput.* 70 (2018) 11–20.
- [46] W. Li, R. Zhao, T. Xiao, X. Wang, Deepreid: deep filter pairing neural network for person re-identification, *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 152–159.
- [47] X. Han, T. Leung, Y. Jia, R. Sukthankar, A.C. Berg, Matchnet: Unifying feature and metric learning for patch-based matching, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286.
- [48] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning, *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.
- [49] A. Santoro, D. Raposo, D.G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, T. Lillicrap, A simple neural network module for relational reasoning, *Advances in Neural Information Processing Systems*, 2017, pp. 4967–4976.
- [50] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, *Computer Vision and Pattern Recognition*, 2018.
- [51] Y. Gavet, M. Fernandes, J. Debayle, J.-C. Pinoli, Dissimilarity criteria and their comparison for quantitative evaluation of image segmentation: application to human retina vessels, *Mach. Vis. Appl.* 25 (8) (2014) 1953–1966.
- [52] X. Tan, S. Chen, J. Li, Z.-H. Zhou, Learning non-metric partial similarity based on maximal margin criterion, *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 138–145.
- [53] A. Tversky, I. Gati, Similarity, separability, and the triangle inequality, *Psychol. Rev.* 89 (2) (1982) 123.
- [54] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [55] P. Bhagat, P. Choudhary, Image annotation: then and now, *Image Vis. Comput.* 80 (2018) 1–23.
- [56] Y. Movshovitz-Attias, A. Toshev, T.K. Leung, S. Ioffe, S. Singh, No fuss distance metric learning using proxies, *IEEE International Conference on Computer Vision*, IEEE, 2017, pp. 360–368.
- [57] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, *Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.
- [58] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: improving particular object retrieval in large scale image databases, *Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.
- [59] B. Kulis, I. Gati, Metric learning: a survey, *Found. Trends® Mach. Learn.* 5 (4) (2013) 287–364.
- [60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [61] Y. Xu, Y. Han, R. Hong, Q. Tian, Sequential video VLAD: training the aggregation locally and temporally, *IEEE Trans. Image Process.* 27 (10) (2018) 4933–4944.
- [62] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to End Learning for Self-driving Cars, *arXiv preprint arXiv:1604.07316*, 2016.